
Программирование на языке Паскаль

Сортировка массивов

Сортировка

Сортировка – это расстановка элементов массива в заданном порядке (по возрастанию, убыванию, последней цифре, сумме делителей, ...).

Задача: переставить элементы массива в порядке возрастания.

Алгоритмы:

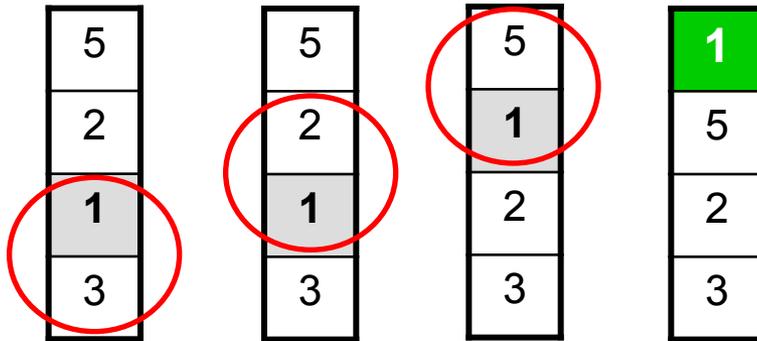
- сортировка обменом – «пузырьковая»
- сортировка выбором
- сортировка вставками
- сортировка подсчетом

Метод пузырька

Идея – пузырек воздуха в стакане воды поднимается со дна вверх.

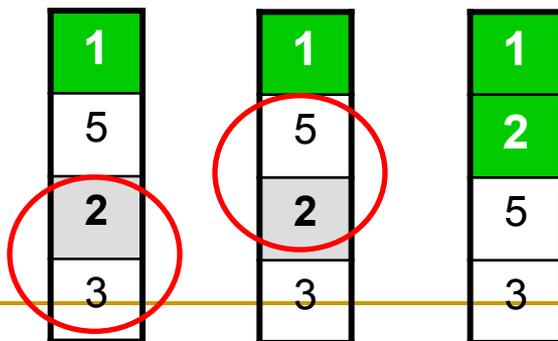
Для массивов – самый маленький ("легкий") элемент перемещается вверх ("всплывает").

1-ый проход

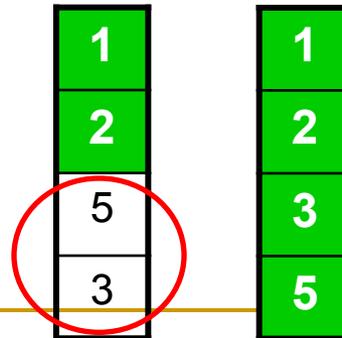


- начиная снизу, сравниваем два соседних элемента; если они стоят "неправильно", меняем их местами
- за 1 проход по массиву **один** элемент (самый маленький) становится на свое место

2-ой проход



3-ий проход



Для сортировки массива из N элементов нужен N-1 проход (достаточно поставить на свои места N-1 элементов).



```
Program C;  
uses crt;      {пузырьковая сортировка}  
var a:array [1..30] of integer;  
i,d,l:integer;  
begin  
  clrscr;  
  randomize;  
  writeln ('исходный массив');  
  for i:= 1 to 30 do  
    begin  
      a[i]:=random(10);  
      write (a[i], ' ');  
    end;  
  writeln;  
  for l:=30 downto 2 do  
    for i:=1 to l-1 do  
      if a[i]>a[i+1] then  
        begin  
          d:=a[i];  
          a[i]:=a[i+1];  
          a[i+1]:=d;  
        end;  
  writeln('новый отсортированный массив');  
  for i:=1 to 30 do  
    write (a[i], ' ');  
readkey;  
end.
```

Программная реализация алгоритма

ИСХОДНЫЙ МАССИВ

8 5 7 8 0 7 2 4 8 6 8 7 0 3 9 3 1 4 2 5 4 2 0 1 8 2 4 8 3 3

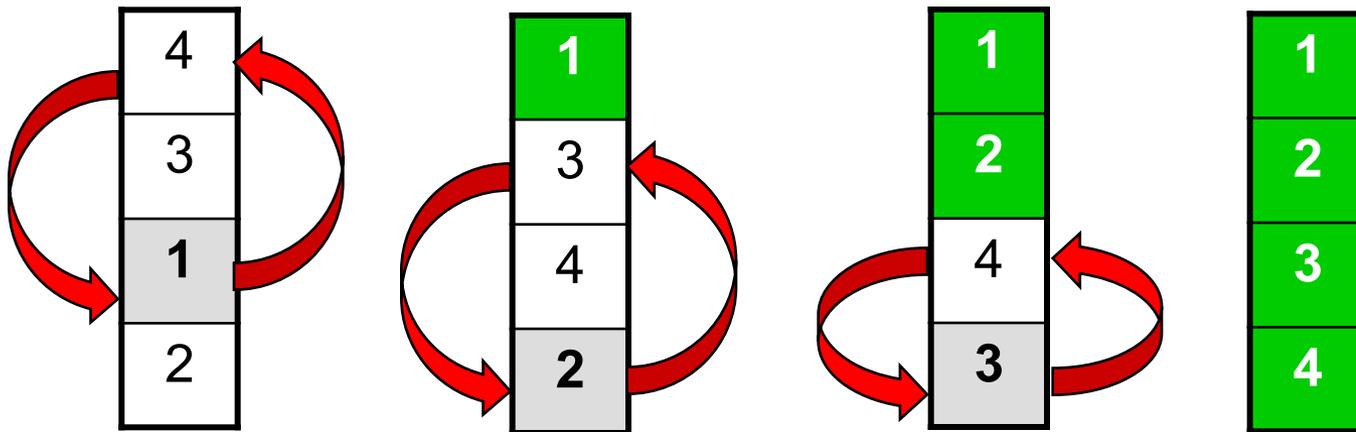
НОВЫЙ ОТСОРТИРОВАННЫЙ МАССИВ

0 0 0 1 1 2 2 2 2 3 3 3 3 4 4 4 4 5 5 6 7 7 7 8 8 8 8 8 8 9

Метод выбора

Идея:

- найти минимальный элемент и поставить на первое место (поменять местами с $A[1]$)
- из оставшихся найти минимальный элемент и поставить на второе место (поменять местами с $A[2]$), и т.д.





Программная реализация алгоритма

```
Program C;      {сортировка выбором}
uses crt;
var b,a:array [1..30] of integer;
i,h,k,d,l:integer;
begin
  clrscr;
  randomize;
  writeln ('исходный массив');
  for i:= 1 to 30 do
    begin
      a[i]:=random(10);
      write (a[i], ' ');
    end;
  writeln;
  for l:=1 to 29 do
    begin
      k:=30-l+1;
      h:=k;
      for i:=1 to 30-l do
        if (a[i]>a[h]) then h:=i;
      d:=a[k]; a[k]:=a[h]; a[h]:=d;
    end;
  writeln('новый отсортированный массив');
  for i:=1 to 30 do
    write (a[i], ' ');
  readkey;
end.
```

ИСХОДНЫЙ МАССИВ

0 7 3 8 7 2 4 8 4 7 3 0 8 5 4 7 8 4 2 7 1 1 0 8 8 5 8 3 1 7

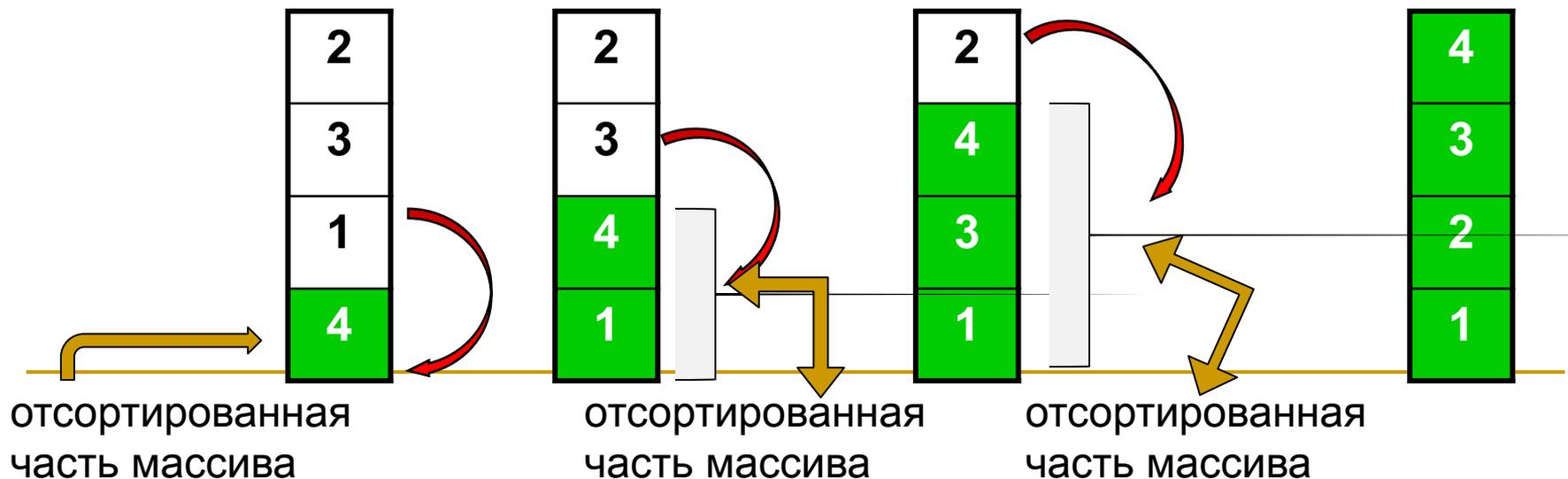
НОВЫЙ ОТСОРТИРОВАННЫЙ МАССИВ

0 0 0 1 1 1 2 2 3 3 3 4 4 4 4 5 5 7 7 7 7 7 8 8 8 8 8 8 8

Сортировка вставками

Идея:

- основана на внедрении в отсортированную часть массива элемента следующего за этой частью, если он удовлетворяет условию сортировки.
- на первом шаге сортировки второй элемент сравнивается с первым, на втором шаге третий элемент сравнивается с двумя первыми и т. д.
- среди уже отсортированных $i-1$ элементов массива вставляют i -й элемент без нарушения порядка, т. е. при вставке i -го элемента на j -е место ($j < i$) элементы с индексами $> j$ и $< i$ увеличивают свой номер на единицу.





Программная реализация алгоритма

```
Program C; {сортировка вставками}
uses crt;
var a:array [1..30] of integer;
i,h,d,l:integer;
begin
  clrscr;
  randomize;
  writeln ('исходный массив');
  for i:= 1 to 30 do
    begin
      a[i]:=random(10);
      write (a[i], ' ');
    end;
  writeln;
  for l:=2 to 30 do
  begin
    d:=a[l]; h:=1;
    while d>a[h] do h:=h+1;
    for i:=l downto h+1 do
      a[i]:=a[i-1];
    a[h]:=d;
  end;
  writeln('новый отсортированный массив');
  for i:=1 to 30 do
    write (a[i], ' ');
  readkey;
end.
```

ИСХОДНЫЙ МАССИВ

6 2 4 8 1 2 4 3 9 8 2 1 3 1 6 8 2 5 8 4 1 1 2 7 5 3 4 3 6 1

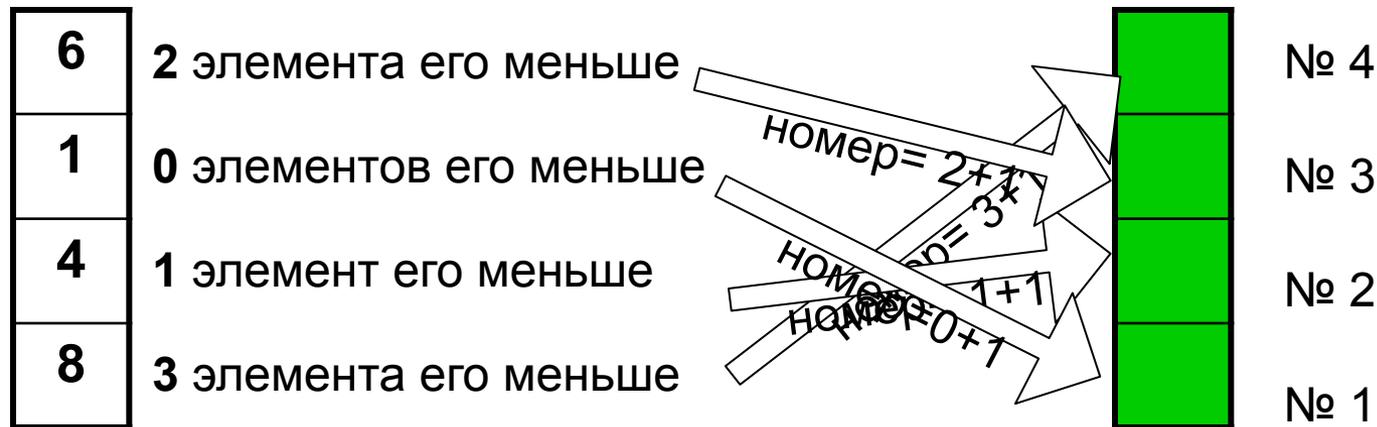
НОВЫЙ ОТСОРТИРОВАННЫЙ МАССИВ

1 1 1 1 1 1 2 2 2 2 2 3 3 3 3 4 4 4 4 5 5 6 6 6 7 8 8 8 8 9

Сортировка подсчетом

Идея:

- основана на подсчете для каждого элемента количества элементов массива, меньших данному.
- от этого количества зависит номер каждого элемента в новом массиве, т. е. если 5 элементов меньше данного, то его место в новом массиве будет 6-ым (при сортировке по возрастанию).



Массив А

Массив В



Программная

реализация алгоритма

```
Program C; {сортировка подсчетом}
uses crt;
var b,a:array [1..30] of integer;
i,h,l:integer;
begin
  clrscr;
  randomize;
  writeln ('исходный массив');
  for i:= 1 to 30 do
    begin
      a[i]:=random(10);
      write (a[i], ' ');
    end;
  writeln;
  for l:= 1 to 30 do
    begin
      h:=0;
      for i:= 1 to 30 do
        if (a[i]<a[l]) and (i<>l) then h:=h+1;
        b[h+1]:=a[l];
      end;
    end;
  for i:=2 to 30 do
    if b[i]=0 then b[i]:=b[i-1];
  writeln('новый массив');
  for i:=1 to 30 do write (b[i], ' ');
  readkey; end.
```

ИСХОДНЫЙ МАССИВ

6 3 7 7 9 7 7 2 3 3 7 3 2 0 8 2 6 4 6 4 6 8 2 5 5 2 6 5 8 1

НОВЫЙ ОТСОРТИРОВАННЫЙ МАССИВ

0 1 2 2 2 2 2 3 3 3 3 4 4 5 5 5 6 6 6 6 6 7 7 7 7 7 8 8 8 9

Задания



"5": Заполнить массив из 10 элементов случайными числами в интервале [0..100] и отсортировать его по последней цифре.

Пример:

Исходный массив:

14 25 13 30 76 58 32 11 41 97

Результат:

30 11 41 32 13 14 25 76 97 58



решение задачи

"4": Заполнить массив из 10 элементов случайными числами в интервале [0..100] и отсортировать первую половину по возрастанию, а вторую – по убыванию.

Пример:

Исходный массив:

14 25 13 30 76 58 32 11 41 97

Результат:

13 14 25 30 76 | 97 58 41 32 11



решение задачи

```
Program gr3;
uses crt;      {используем пузырьковую сортировку}
var d:array [1..10] of byte;
i,k,h:byte;
begin
  clrscr;
  randomize;
  writeln ('исходный массив');
  for i:= 1 to 10 do
    begin
      d[i]:=random(101);
      write (d[i], ' ');
    end;
  writeln;
  for k:=5 downto 2 do
    for i:=1 to k-1 do
      if d[i]>d[i+1] then
        begin
          h:=d[i]; d[i]:=d[i+1]; d[i+1]:=h;
        end;
  for k:=10 downto 7 do
    for i:=6 to k-1 do
      if d[i]<d[i+1] then
        begin
          h:=d[i]; d[i]:=d[i+1]; d[i+1]:=h;
        end;
  writeln('полученный массив');
for i:=1 to 10 do
  write (d[i], ' ');
Readkey; end.
```

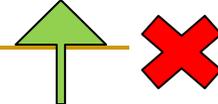
Результат работы группы № 3

ИСХОДНЫЙ МАССИВ

26 11 23 76 62 79 88 18 55 100

ПОЛУЧЕННЫЙ МАССИВ

11 23 26 62 76 100 88 79 55 18



```
Program gr2;
uses crt; {используем пузырьковую сортировку}
var w:array [1..10] of byte;
i,k,h:byte;
begin
  clrscr;
  randomize;
  writeln ('исходный массив');
  for i:= 1 to 10 do
    begin
      w[i]:=random(101);
      write (w[i], ' ');
    end;
  writeln;
  for k:=10 downto 2 do
    for i:=1 to k-1 do
      if w[i] mod 10 > w[i+1] mod 10 then
        begin
          h:=w[i];
          w[i]:=w[i+1];
          w[i+1]:=h;
        end;
    writeln('полученный массив');
  for i:=1 to 10 do
    write (w[i], ' ');

Readkey;
end.
```

Результат работы группы № 2

ИСХОДНЫЙ МАССИВ

86 5 20 32 88 15 17 14 34 99

ПОЛУЧЕННЫЙ МАССИВ

20 32 14 34 5 15 86 17 88 99